

artdaq Utilities - Bug #22146

DAQInterface should handle SIGHUP, SIGTERM, etc. as gracefully as possible

03/15/2019 11:26 AM - John Freeman

Status:	Reviewed	Start date:	03/15/2019
Priority:	Normal	Due date:	
Assignee:	John Freeman	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Experiment:	-	Co-Assignees:	Eric Flumerfelt

Description

This Issue is motivated by Eric's Issue [#22095](#), in which he observed that sometimes when using run_demo.sh not all artdaq processes (especially datalogger) appeared to get cleaned up. Perhaps related, running the demo config with component01 and component02 on woof, what I've found is that when it's in the running state, whether DAQInterface is controlling processes in "pmt" or "direct" mode, if it receives a SIGHUP or a SIGTERM then while the python script that is DAQInterface disappears, the artdaq processes (and in the case of "pmt" mode, pmt.rb) remain. While it's the case that if you relaunch and then try to run again with the same processes, DAQInterface will clean up the processes after complaining and then put itself back in the "stopped" state, there's of course no guarantee in the real world that this action will be taken subsequent to an unexpected DAQInterface killing. The possibility of DAQInterface catching kill signals and then gracefully winding down active artdaq processes should be investigated.

Associated revisions

Revision 3d6d95ef - 03/25/2019 05:19 PM - John Freeman

JCF: improve the logic used to ensure that daqinterface.py itself dies after it's cleaned everything up after catching a signal (see Issue #22146 comments from today)

Revision d5dd76db - 08/07/2019 03:39 PM - John Freeman

JCF: implement Ron's suggested change so that DAQInterface aliases to an executable script, not a sourced script

As you can see from the diff, this is a very simple change. I've performed a few regression tests and confirmed the following remains the case:

-You can run two DAQInterface instances in the background at the same time on separate partitions in the same terminal, using the DAQINTERFACE_PARTITION_NUMBER environment variable to control which one you're sending transitions to.

-If you close the terminal DAQInterface is running in, or hit Ctrl-c on it (if it's running in the foreground), then the root file closes correctly (Issue #22146)

-Output goes simultaneously to the screen and to the file referred to by \$DAQINTERFACE_LOGFILE.

History

#1 - 03/23/2019 11:00 PM - John Freeman

- % Done changed from 0 to 100
- Status changed from New to Resolved

Resolved at the head of the feature/issue22146_handle_signals branch, commit a9bbd7dcaa1a27c09798f88ef4d0c1a7f82b9576.

DAQInterface will now enter the recover transition (i.e., sending a stop and then a shutdown to artdaq processes found in the running state before killing them, most notably resulting in a correctly-saved root file) if it receives any of the following signals:

- SIGINT, meaning that if DAQInterface is running in the foreground in a terminal and you hit Ctrl-c
- SIGHUP, meaning you close the terminal DAQInterface is running in
- SIGTERM, meaning you kill DAQInterface (by ignoring the are-you-sure warning you get when DAQInterface isn't in the "stopped" state but you try killing it via the kill_daqinterface_on_partition.sh script

#2 - 03/25/2019 11:42 AM - Eric Flumerfelt

I've noticed a few cases where closing the DAQInterface window has led to a python process remaining active, with

/tmp/daqitnerface-\$USER/DAQInterface_partition*.log showing no activity. I'm not sure what the workaround might be, other than making sure to proceed with default handlers after running the DAQInterface signal handler...

```
def_term_handler = signal.SIG_DFL
def_hup_handler = signal.SIG_DFL
def_int_handler = signal.SIG_DFL
```

...

```
--- sys.exit(1)
++ if signum == signal.SIGTERM
++   def_term_handler(signum, stack)
+++ else if signum ...
```

...

```
def_term_handler = signal.signal(signal.SIGTERM, handle_kill_signal)
```

...

#3 - 03/25/2019 05:23 PM - John Freeman

To address Eric's findings, with commit 3d6d95ef1a9b10169285b8bab25de68f2e024752 on feature/issue22146_handle_signals, after putting itself through the recover transition, DAQInterface will then call the default signal handler, and then as an insurance policy call `os._exit`, which is a harder exit than `sys.exit`.

#4 - 03/28/2019 01:39 PM - Eric Flumerfelt

- Status changed from *Resolved* to *Reviewed*
- Co-Assignees *Eric Flumerfelt* added

I've tried closing the window at several mid-transition and between transitions, and now no longer see the issue. Code review looks good. Merged into develop.